Network Working Group
Request for Comments: 4383
Category: Standards Track

M. Baugher
Cisco
E. Carrara
Royal Institute of Technology

February 2006

The Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Secure Real-time Transport Protocol (SRTP)

Status of This Memo

Copyright Notice

Abstract

This memo describes the use of the Timed Efficient Stream Loss-tolerant Authentication (RFC 4082) transform within the Secure Real-time Transport Protocol (SRTP), to provide data origin authentication for multicast and broadcast data streams.

Table of Contents

1. Introduction

Multicast and broadcast communications introduce some new security challenges compared to unicast communication. Many multicast and broadcast applications need "data origin authentication" (DOA), or "source authentication", in order to guarantee that a received message had originated from a given source, and was not manipulated during the transmission. In unicast communication, a pairwise security association between one sender and one receiver can provide data origin authentication using symmetric-key cryptography (such as a message authentication code, MAC). When the communication is strictly pairwise, the sender and receiver agree upon a key that is known only to them.

In groups, however, a key is shared among more than two members, and this symmetric-key approach does not guarantee data origin authentication. When there is a group security association [RFC4046] instead of a pairwise security association, any of the members can alter the packet and impersonate any other member. The MAC in this case only guarantees that the packet was not manipulated by an attacker outside the group (and hence not in possession of the group key), and that the packet was sent by a source within the group.

Some applications cannot tolerate source ambiguity and need to identify the true sender from any other group member. A common way to solve the problem is by use of asymmetric cryptography, such as digital signatures. This method, unfortunately, suffers from high overhead in terms of time (to sign and verify) and bandwidth (to convey the signature in the packet).

Several schemes have been proposed to provide efficient data origin authentication in multicast and broadcast scenarios. The Timed Efficient Stream Loss-tolerant Authentication (TESLA) is one such scheme.

This memo specifies TESLA authentication for SRTP. SRTP TESLA can provide data origin authentication to RTP applications that use group security associations (such as multicast RTP applications) so long as receivers abide by the TESLA security invariants [RFC4082].

1.1. Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification assumes that the reader is familiar with both SRTP and TESLA. Few of their details are explained in this document, and the reader can find them in their respective specifications, [RFC3711] and [RFC4082]. This specification uses the same definitions as TESLA for common terms and assumes that the reader is familiar with the TESLA algorithms and protocols [RFC4082].

2. SRTP

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of RTP, which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the RTP control protocol, the Real-time Transport Control Protocol (RTCP). Note that the term "SRTP" may often be used to indicate SRTCP as well.

SRTP is a framework that allows new security functions and new transforms to be added. SRTP currently does not define any mechanism to provide data origin authentication for group security associations. Fortunately, it is straightforward to add TESLA to the SRTP cryptographic framework.

The TESLA extension to SRTP is defined in this specification, which assumes that the reader is familiar with the SRTP specification [RFC3711], its packet structure, and its processing rules. TESLA is

an alternative message-authentication algorithm that authenticates messages from the source when a key is shared among two or more receivers.

3. TESLA

TESLA provides delayed per-packet data authentication and is specified in [RFC4082].

In addition to its SRTP data-packet definition given here, TESLA needs an initial synchronization protocol and initial bootstrapping procedure. The synchronization protocol allows the sender and the receiver to compare their clocks and determine an upper bound of the difference. The synchronization protocol is outside the scope of this document.

TESLA also requires an initial bootstrapping procedure to exchange needed parameters and the initial commitment to the key chain [RFC4082]. For SRTP, it is assumed that the bootstrapping is performed out-of-band, possibly using the key management protocol that is exchanging the security parameters for SRTP, e.g., [RFC3547, RFC3830]. Initial bootstrapping of TESLA is outside the scope of this document.

4. Usage of TESLA within SRTP

The present specification is an extension to the SRTP specification [RFC3711] and describes the use of TESLA with only a single key chain and delayed-authentication [RFC4082].

4.1. The TESLA Extension

TESLA is an OPTIONAL authentication transform for SRTP. When used, TESLA adds the fields shown in Figure 1 per-packet. The fields added by TESLA are called "TESLA authentication extensions," whereas "authentication tag" or "integrity protection tag" indicate the normal SRTP integrity protection tag, when the SRTP master key is shared by more than two endpoints [RFC3711].

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               i                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Disclosed Key                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            TESLA MAC                          ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. The "TESLA authentication extension".

i: 32 bit, MANDATORY
   Identifier of the time interval i, corresponding to the key K_i, which is used to calculate the TESLA MAC of the current packet (and other packets sent in the current time interval i).

Disclosed Key: variable length, MANDATORY
   The disclosed key (K_(i-d)), which can be used to authenticate previous packets from earlier time intervals [RFC4082]. A Section 4.3 parameter establishes the size of this field.

TESLA MAC (Message Authentication Code): variable length, MANDATORY
   The MAC computed using the key K'_i (derived from K_i) [RFC4082], which is disclosed in a subsequent packet (in the Disclosed Key field). The MAC coverage is defined in Section 4.6. A Section 4.3 parameter establishes the size of this field.

4.2. SRTP Packet Format

Figure 2 illustrates the format of the SRTP packet when TESLA is applied. When applied to RTP, the TESLA authentication extension SHALL be inserted before the (optional) SRTP MKI and (recommended) authentication tag (SRTP MAC).

```
     0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<+<+
  |V=2|P|X|  CC   |M|     PT      |       sequence number         | | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
  |                           timestamp                           | | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
  |           synchronization source (SSRC) identifier            | | |
  +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+ | |
  |            contributing source (CSRC) identifiers             | | |
  |                               ....                            | | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
  |                   RTP extension (OPTIONAL)                    | | |
+>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| |                          payload  ...                         | | |
| |                               +-------------------------------+ | |
| |                               | RTP padding   | RTP pad count | | |
+>+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<+ |
| |                            i                                  | | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                     Disclosed Key                             ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                         TESLA MAC                             ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<|-+
| ~                            MKI                                ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                            MAC                                ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
|                                                                   | |
+- Encrypted Portion                 TESLA Authenticated Portion ---+ |
                                                                      |
                                             Authenticated Portion ---+
```

Figure 2. The format of the SRTP packet when TESLA is applied.

As in SRTP, the "Encrypted Portion" of an SRTP packet consists of the encryption of the RTP payload (including RTP padding when present) of the equivalent RTP packet.

The "Authenticated Portion" of an SRTP packet consists of the RTP header, the Encrypted Portion of the SRTP packet, and the TESLA authentication extension. Note that the definition is extended from [RFC3711] by the inclusion of the TESLA authentication extension.

The "TESLA Authenticated Portion" of an SRTP packet consists of the RTP header and the Encrypted Portion of the SRTP packet. As shown in Figure 2, the SRTP MAC covers up to the MKI field but does not include the MKI. It is necessary for packet integrity that the SRTP-TESLA MAC tag be covered by the SRTP integrity check. SRTP does not cover the MKI field (because it does not need to be covered for SRTP packet integrity). In order to make the two tags (SRTP-TESLA MAC and SRTP-MAC) contiguous, we would need to redefine the SRTP specification to include the MKI in SRTP-MAC coverage. This change is impossible, so the MKI field separates the TESLA MAC from the SRTP MAC in the packet layout of Figure 2. This change to the packet format presents no problem to an implementation that supports the new SRTP-TESLA authentication transform.

The lengths of the Disclosed Key and TESLA MAC fields are Section 4.3 parameters. As in SRTP, fields that follow the packet payload are not necessarily aligned on 32-bit boundaries.

4.3. Extension of the SRTP Cryptographic Context

When TESLA is used, the definition of cryptographic context in Section 3.2 of SRTP SHALL include the following extensions.

Transform-Dependent Parameters

1. an identifier for the PRF (TESLA PRF), implementing the one-way function F(x) in TESLA (to derive the keys in the chain), and the one-way function F'(x) in TESLA (to derive the keys for the TESLA MAC, from the keys in the chain), e.g., to indicate HMAC-SHA1. See Section 6 for the default value.

2. a non-negative integer, n_p, determining the length of the F output; i.e., the length of the keys in the chain (that is also the key disclosed in an SRTP packet). See Section 6 for the default value.

3. a non-negative integer, n_f, determining the length of the output of F', i.e., of the key for the TESLA MAC. See Section 6 for the default value.

4. an identifier for the TESLA MAC that accepts the output of F'(x) as its key, e.g., to indicate HMAC-SHA1. See Section 6 for the default value.

5. a non-negative integer, n_m, determining the length of the output of the TESLA MAC. See Section 6 for the default value.

6. the beginning of the session T_0.

7. the interval duration T_int (in msec).

8. the key disclosure delay d (in number of intervals).

9. the upper bound D_t (in sec) on the lag of the receiver clock relative to the sender clock (this quantity has to be calculated by the peers out-of-band).

10. a non-negative integer, n_c, determining the length of the key chain, K_0...K_n-1 of [RFC4082] (see also Section 6 of this document), which is determined based upon the expected duration of the stream.

11. the initial key of the chain to which the sender has committed himself.

F(x) is used to compute a keychain of keys in SRTP TESLA, as defined in Section 6. Also according to TESLA, F'(x) computes a TESLA MAC key with inputs as defined in Section 6.

Section 6 of this document defines the default values for the transform-specific TESLA parameters.

4.4. SRTP Processing

The SRTP packet processing is described in Section 3.3 of the SRTP specification [RFC3711]. The use of TESLA slightly changes the processing, as the SRTP MAC is checked upon packet arrival for DoS prevention, but the current packet is not TESLA-authenticated. Each packet is buffered until a subsequent packet discloses its TESLA key. The TESLA verification itself consists of some steps, such as tests of TESLA security invariants, that are described in Sections 3.5-3.7 of [RFC4082]. The words "TESLA computation" and "TESLA verification" hereby imply all those steps, which are not all spelled out in the following. In particular, notice that the TESLA verification implies checking the safety condition (Section 3.5 of [RFC4082]).

As pointed out in [RFC4082], if the packet is deemed "unsafe", then the receiver considers the packet unauthenticated. It should discard unsafe packets, but, at its own risk, it may choose to use them unverified. Hence, if the safe condition does not hold, it is RECOMMENDED to discard the packet and log the event.

4.4.1. Sender Processing

The sender processing is as described in Section 3.3 of [RFC3711], up to step 5, inclusive. After that, the following process is followed:

6. When TESLA is applied, identify the key in the TESLA chain to be used in the current time interval, and the TESLA MAC key derived from it. Execute the TESLA computation to obtain the TESLA authentication extension for the current packet, by appending the current interval identifier (as i field), the disclosed key of the chain for the previous disclosure interval (i.e., the key for interval i is disclosed in interval i+d), and the TESLA MAC under the current key from the chain. This step uses the related TESLA parameters from the crypto context as for Step 4.

7. If the MKI indicator in the SRTP crypto context is set to one, append the MKI to the packet.

8. When TESLA is applied, and if the SRTP authentication (external tag) is required (for DoS), compute the authentication tag as described in step 7 of Section 3.3 of the SRTP specification, but with coverage as defined in this specification (see Section 4.6).

9. If necessary, update the rollover counter (step 8 in Section 3.3 of [RFC3711]).

4.4.2. Receiver Processing

The receiver processing is as described in Section 3.3 of [RFC3711], up to step 4, inclusive.

To authenticate and replay-protect the current packet, the processing is as follows:

First, check if the packet has been replayed (as per Section 3.3 of [RFC3711]). Note, however, that the SRTP replay list contains SRTP indices of recently received packets that have been authenticated by TESLA (i.e., replay list updates MUST NOT be based on SRTP MAC). If the packet is judged to be replayed, then the packet MUST be discarded, and the event SHOULD be logged.

Next, perform verification of the SRTP integrity protection tag (not the TESLA MAC), if present, using the rollover counter from the current packet, the authentication algorithm indicated in the cryptographic context, and the session authentication key. If the verification is unsuccessful, the packet MUST be discarded from further processing, and the event SHOULD be logged.

If the verification is successful, remove and store the MKI (if present) and authentication tag fields from the packet. The packet is buffered, awaiting disclosure of the TESLA key in a subsequent packet.

TESLA authentication is performed on a packet when the key is disclosed in a subsequent packet. Recall that a key for interval i is disclosed during interval i+d, i.e., the same key is disclosed in packets sent over d intervals of length t_int. If the interval identifier i from the packet (Section 4.1) has advanced more than d intervals from the highest value of i that has been received, then packets have been lost, and one or more keys MUST be computed as described in Section 3.2, second paragraph, of the TESLA specification [RFC4082]. The computation is performed recursively for all disclosed keys that have been lost, from the newly-received interval to the last-received interval.

When a newly-disclosed key is received or computed, perform the TESLA verification of the packet using the rollover counter from the packet, the TESLA security parameters from the cryptographic context, and the disclosed key. If the verification is unsuccessful, the packet MUST be discarded from further processing, and the event SHOULD be logged. If the TESLA verification is successful, remove the TESLA authentication extension from the packet.

To decrypt the current packet, the processing is as follows:

Decrypt the Encrypted Portion of the packet, using the decryption algorithm indicated in the cryptographic context, the session encryption key, and salt (if used) found in Step 4 with the index from Step 2.

(Note that the order of decryption and TESLA verification is not mandated. It is RECOMMENDED that the TESLA verification be performed before decryption. TESLA application designers might choose to implement optimistic processing techniques such as notification of TESLA verification results after decryption or even after plaintext processing. Optimistic verification is beyond the scope of this document.)

Update the rollover counter and highest sequence number, s_l, in the cryptographic context, using the packet index estimated in Step 2. If replay protection is provided, also update the Replay List (i.e., the Replay List is updated after the TESLA authentication is successfully verified).

4.5. SRTCP Packet Format

Figure 3 illustrates the format of the SRTCP packet when TESLA is applied. The TESLA authentication extension SHALL be inserted before the MKI and authentication tag. Recall from [RFC3711] that in SRTCP the MKI is OPTIONAL, while the E-bit, the SRTCP index, and the authentication tag are MANDATORY. This means that the SRTP (external) MAC is MANDATORY also when TESLA is used.

As in SRTP, the "Encrypted Portion" of an SRTCP packet consists of the encryption of the RTCP payload of the equivalent compound RTCP packet, from the first RTCP packet, i.e., from the ninth (9) byte to the end of the compound packet.

The "Authenticated Portion" of an SRTCP packet consists of the entire equivalent (eventually compound) RTCP packet, the E flag, the SRTCP index (after any encryption has been applied to the payload), and the TESLA extension. Note that the definition is extended from [RFC3711] by the inclusion of the TESLA authentication extension.

We define the "TESLA Authenticated Portion" of an SRTCP packet as consisting of the RTCP header (first 8 bytes) and the Encrypted Portion of the SRTCP packet.

Processing of an SRTCP packets is similar to the SRTP processing (Section 4.3), but there are SRTCP-specific changes described in Section 3.4 of the SRTP specification [RFC3711] and in Section 4.6 of this memo.

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<+<+
  |V=2|P|    RC   |   PT=SR or RR   |             length          | | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
  |                         SSRC of sender                        | | |
+>+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+ | |
| ~                          sender info                          ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                         report block 1                        ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                         report block 2                        ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                              ...                              ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| |V=2|P|    SC   |  PT=SDES=202  |             length            | | |
| +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+ | |
| |                          SSRC/CSRC_1                          | | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                           SDES items                          ~ | |
| +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+ | |
| ~                              ...                              ~ | |
+>+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+ | |
| |E|                         SRTCP index                         | | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<+ |
| |                              i                                | | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                         Disclosed Key                         ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| ~                           TESLA MAC                           ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<|-+
| ~                           SRTCP MKI                           ~ | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
| :                       authentication tag                      : | |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | |
|                                                                   | |
+-- Encrypted Portion              TESLA Authenticated Portion -----+ |
                                                                      |
                                         Authenticated Portion -------+
```

Figure 3. The format of the SRTCP packet when TESLA is applied.

Note that when additional fields are added to a packet, it will increase the packet size and thus the RTCP average packet size.

4.6. TESLA MAC

Let M’ denote packet data to be TESLA-authenticated. In the case of SRTP, M’ SHALL consist of the SRTP TESLA Authenticated Portion (RTP header and SRTP Encrypted Portion; see Figure 2) of the packet concatenated with the rollover counter (ROC) of the same packet:

M’ = ROC || TESLA Authenticated Portion.

In the case of SRTCP, M’ SHALL consist of the SRTCP TESLA Authenticated Portion only (RTCP header and SRTCP Encrypted Portion).

The normal authentication tag (OPTIONAL for SRTP, MANDATORY for SRTCP) SHALL be applied with the same coverage as specified in [RFC3711]. That is:

- for SRTP: Authenticated Portion || ROC (with the extended definition of SRTP Authentication Portion as in Section 4.2).

- for SRTCP: Authenticated Portion (with the extended definition of SRTCP Authentication Portion as in Section 4.2).

The predefined authentication transform in SRTP, HMAC-SHA1 [RFC2104], is also used to generate the TESLA MAC. For SRTP (and respectively for SRTCP), the HMAC SHALL be applied to the key in the TESLA chain corresponding to a particular time interval, and to M’ as specified above. The HMAC output SHALL then be truncated to the n_m left-most bits. Default values are in Section 6.

As with SRTP, the predefined HMAC-SHA1 authentication algorithm MAY be replaced with an alternative algorithm that is specified in a future Internet RFC.

4.7. PRFs

TESLA requires a pseudo-random function (PRF) to implement

* one one-way function F(x) to derive the key chain, and
* one one-way function F’(x) to derive (from each key of the chain) the key that is actually used to calculate the TESLA MAC.

When TESLA is used within SRTP, the default choice of the PRF SHALL be HMAC-SHA1. Default values are in Section 6.

Other PRFs can be chosen, and their use SHALL follow the common guidelines in [RFC3711] when adding new security parameters.

5. TESLA Bootstrapping and Cleanup

The extensions to the SRTP cryptographic context include a set of TESLA parameters that are listed in Section 4.3 of this document. Furthermore, TESLA MUST be bootstrapped at session setup (for the parameter exchange and the initial key commitment) through a regular data authentication system (a digital signature algorithm is RECOMMENDED). Key management procedures can take care of this bootstrapping prior to the commencement of an SRTP session where TESLA authentication is used. The bootstrapping mechanism is out of scope for this document (it could, for example, be part of the key management protocol).

A critical factor for the security of TESLA is that the sender and receiver need to be loosely synchronized. TESLA requires a bound on clock drift to be known (D_t). Use of TESLA in SRTP assumes that the time synchronization is guaranteed by out-of-band schemes (e.g., key management). That is, it is not in the scope of SRTP.

It also should be noted that TESLA has some reliability requirements in that a key is disclosed for a packet in a subsequent packet, which can get lost. Since a key in a lost packet can be derived from a future packet, TESLA is robust to packet loss. This key stream stops, however, when the key-bearing data stream packets stop at the conclusion of the RTP session. To avoid this nasty boundary condition, send null packets with TESLA keys for one entire key-disclosure period following the interval in which the stream ceases: Null packets SHOULD be sent for d intervals of duration t_int (items 8 and 9 of Section 4.3). The rate of null packets SHOULD be the average rate of the session media stream.

6. SRTP TESLA Default Parameters

Key management procedures establish SRTP TESLA operating parameters, which are listed in Section 4.3 of this document. The operating parameters appear in the SRTP cryptographic context and have the default values that are described in this section. In the future, an Internet RFC MAY define alternative settings for SRTP TESLA that are different than those specified here. In particular, note that the settings defined in this memo can have a large impact on bandwidth, as they add 38 bytes to each packet (when the field length values are the default ones). For certain applications, this overhead may represent more than a 50% increase in packet size. Alternative settings might seek to reduce the number and length of various TESLA fields and outputs. No such optimizations are considered in this memo.

It is RECOMMENDED that the SRTP MAC be truncated to 32 bits, since the SRTP MAC provides only group authentication and serves only as protection against external DoS.

The default values for the security parameters are listed in the following table.

Parameter	Mandatory-to-support	Default
TESLA PRF	HMAC-SHA1	HMAC-SHA1
BIT-OUTPUT LENGTH n_p	160	160
BIT-OUTPUT LENGTH n_f	160	160
TESLA MAC	HMAC-SHA1	HMAC-SHA1
(TRUNCATED) BIT-OUTPUT LENGTH n_m	80	80

As shown above, TESLA implementations MUST support HMAC-SHA1 [RFC2104] for the TESLA MAC and the TESLA PRF. The TESLA keychain generator is recursively defined as follows [RFC4082].

$$K_i=HMAC_SHA1(K_\{i+1\},0), i=0..N-1$$

where N-1=n_c from the cryptographic context.

The TESLA MAC key generator is defined as follows [RFC4082].

$$K'_i=HMAC_SHA1(K_i,1)$$

The TESLA MAC uses a truncated output of ten bytes [RFC2104] and is defined as follows.

$$HMAC_SHA1(K'_i, M')$$

where M' is as specified in Section 4.6.

7. Security Considerations

Denial of Service (DoS) attacks on delayed authentication are discussed in [PCST]. TESLA requires receiver buffering before authentication; therefore, the receiver can suffer a denial of service attack due to a flood of bogus packets. To address this problem, the external SRTP MAC, based on the group key, MAY be used in addition to the TESLA MAC. The short size of the SRTP MAC (default 32 bits) is motivated because that MAC is purely for DoS prevention from attackers external to the group. The shorter output tag means that an attacker has a better chance of getting a forged packet accepted, which is about 2^31 attempts on average. As a first line of defense against a denial of service attack, a short tag is

probably adequate; a victim will likely have ample evidence that it is under attack before accepting a forged packet, which will subsequently fail the TESLA check. [RFC4082] describes other mechanisms that can be used to prevent DoS, in place of the external group-key MAC. If used, they need to be added as processing steps (following the guidelines of [RFC4082]).

The use of TESLA in SRTP defined in this specification is subject to the security considerations discussed in the SRTP specification [RFC3711] and in the TESLA specification [RFC4082]. In particular, the TESLA security is dependent on the computation of the "safety condition" as defined in Section 3.5 of [RFC4082].

SRTP TESLA depends on the effective security of the systems that perform bootstrapping (time synchronization) and key management. These systems are external to SRTP and are not considered in this specification.

The length of the TESLA MAC is by default 80 bits. RFC 2104 requires the MAC length to be at least 80 bits and at least half the output size of the underlying hash function. The SHA-1 output size is 160 bits, so both of these requirements are met with the 80-bit MAC specified in this document. Note that IPsec implementations tend to use 96 bits for their MAC values to align the header with a 64-bit boundary. Both MAC sizes are well beyond the reach of current cryptanalytic techniques.

8. Acknowledgements

The authors would like to thank Ran Canetti, Karl Norrman, Mats Naslund, Fredrik Lindholm, David McGrew, and Bob Briscoe for their valuable help.

9. References

9.1. Normative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

9.2. Informative References

[PCST] Perrig, A., Canetti, R., Song, D., Tygar, D., "Efficient and Secure Source Authentication for Multicast", in Proc. of Network and Distributed System Security Symposium NDSS 2001, pp. 35-46, 2001.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.

Authors' Addresses

Questions and comments should be directed to the authors and msec@ietf.org.

Mark Baugher
Cisco Systems, Inc.
5510 SW Orchid Street
Portland, OR 97219 USA

Phone: +1 408-853-4418
EMail: mbaugher@cisco.com

Elisabetta Carrara
Royal Institute of Technology
Stockholm
Sweden

EMail: carrara@kth.se

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).